

Тестовое задание

Данное тестовое задание содержит в себе условия для претендующих на позиции как middle, так и junior. В зависимости от желаемой позиции условия будут различаться. Различающиеся требования будут выделены разными цветами для middle и junior

Цель:

Написать приложение, по управлению складом.

Условия:

- В качестве языка для серверного приложения использовать C#.
- Уровень веб приложения должен быть выполнен на ASP.NET Core.
- Ограничений по выбору ORM нет.
- Клиент должен быть отдельным написан на одном из популярных фреймворков, таких, например, как React/Angular/Vue/Blazor WASM. Для junior разрешается написать клиент на Razor Pages или ASP.NET Core MVC
- В качестве БД необходимо использовать MS SQL Server или PostgreSQL.

Описание задачи:

Необходимо написать приложение, автоматизирующее (упрощенно) работу склада.

Пример выполненного задания: <http://193.32.203.182:8080/>

Для претендента на позицию junior необходимо сделать справочники «Ресурсы» и «Единицы измерения», а также раздел «Поступления».

Для претендующего на позицию middle необходимо реализовать полный функционал.

Набор сущностей:

- Ресурс (идентификатор, наименование, состояние)
- Единица измерения (идентификатор, наименование, состояние)
- Клиент (идентификатор, наименование, адрес, состояние)
- Баланс (идентификатор, идентификатор ресурса, идентификатор единицы измерения, количество)
- Документ поступления (идентификатор, номер, дата)
- Ресурс поступления (идентификатор, идентификатор ресурса, идентификатор единицы измерения, количество)
- Документ отгрузки (идентификатор, номер, идентификатор клиента, дата, состояние)
- Ресурс отгрузки (идентификатор, идентификатор ресурса, идентификатор единицы измерения, количество)

Бизнес-правила:

- Невозможно создать Ресурс, Единицу измерения, Клиента, Документ поступления и Документ отгрузки, если в системе уже существует сущность того же типа с таким наименованием/номером.

- Невозможно удалить Ресурс, Единицу измерения, Клиента, если они где-то используются, но можно перевести их в архив. Сущности в архиве невозможно выбрать при создании/редактировании документов поступления/отгрузки, если только они уже не выбраны в этом документе.
- При создании документа поступления на складе увеличивается баланс, на кол-во поступивших ресурсов. При удалении документа поступления, баланс на складе уменьшается. Так же важно чтобы баланс правильно изменялся при редактировании ресурсов поступления. Если ресурсов на складе недостаточно. То сохранить/удалить документ нельзя.
- При создании/редактировании/удалении документа отгрузки – с балансом ничего не происходит, изменения баланса происходит только когда документ отгрузки подписывается или отзывается. При подписании отгрузки баланс на складе уменьшается, при отзыве – увеличивается. Как и в случае с поступлением, подписать документ можно только в случае, если на складе достаточно ресурсов.
- Баланс «суммируется» на складе, если у него одинаковый ресурс и единица измерения.
- Документ поступления может быть пустым (т.е. не содержать в себе ресурсов), а документ отгрузки не может быть пустым.

Общие требования:

- Необходимо реализовать фильтрацию на страницах Склад, Поступления, Отгрузки. На странице Склад фильтры должны быть по ресурсам и единицам измерения, на страницах Поступления и Отгрузки фильтры должны быть по датам (периоду), по номерам документов, по ресурсам и единицам измерения. Фильтры по номерам документов, ресурсам и единицам измерения должны допускать выбор нескольких элементов (select multiple). Фильтрация должна происходить на сервере. Фильтры по номерам документов, по ресурсам и единицам измерения не должны зависеть от выбранного периода.
- Таблицы по документам поступления/отгрузки должны так же отображать наполнение документа, т.е. сами ресурсы, которые в них входят.
- Приложение должно само создавать базу данных, если она не существует.